A Study of Moore and Mealy FSM Example for Education

Assistant Prof.Tirupati M.Goskula [1], Altamash Ahmed Khan [2], Syed Fahimuddin [2]

[1]Assistant Professor, Electronics & Telecommunication, RTMNU/Anjuman College of Engineering & Tech, Nagpur, Maharashtra, India

[2]Scholars, Electronics & Telecommunication, RTMNU/Anjuman College of Engineering & Tech, Nagpur, Maharashtra, India

***Abstract*—With topical technological advancements, contemporary societies are flattering more and more dependent on the automated machines. Recent automated machines adapt the sequence of their actions subject on their environment and events. The Finite state machine (FSM) is developing those sequences of events or instructions mathematically. In this article, Mealy and Moore, these two FSM are discussed. Different results are demonstrated in categorize to show the significance of FSM modeling. A code detector circuit is planned to implement both Moore and Mealy machines in HDL. It is a FSM design case, can be used for students concepts structure and exhibition. These designs are implemented in HDL. An assessment is also made based on both implementations**.

Index Terms—*FSM; automation; VHDL;, Xilinx-ISE;, timing diagram; computer aided design.(keywords)*

I. Introduction

In any programming languages, there will always be two or more way to code the same problem rather than one. Hardware description languages (HDL) like Verilog and VHDL are no dissimilar. It also delivers several options to the designer as to how to accomplish a same task. So, it is up to the designer to code in an effective way that results in best presentation and minimum source consumption [1]. In this essay, performance evaluation between the Mealy and Moore state instrument is provided. Both has its own advantages and disadvantages so it is up to designer to choose proper design based on application. Same way assessment is done among different State encoding schemes like binary and one encoding. The outcome on area and act by using these schemes is also discussed in the article [1].

II. Type Of State Machines

There are mainly two types of state machine.

1. Mealy state machine
2. Moore state machine

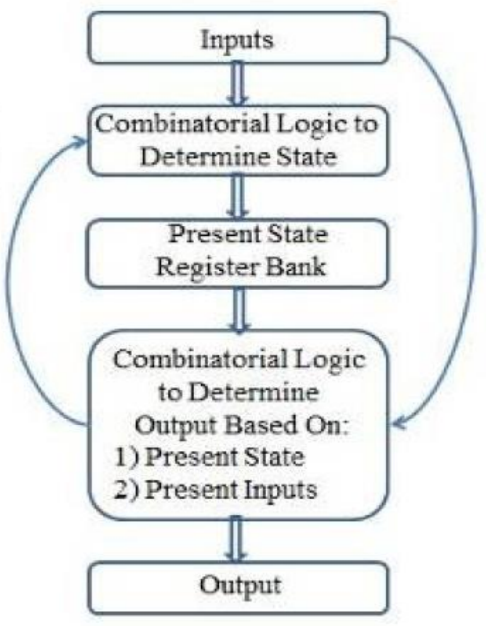
Inputs
Combinatorial Logic to Determine State
Present State Register Bank
Combinatorial Logic to Determine Output Based On: 1) Present State 2) Present Inputs
Output

Fig.1 Mealy State Machine

In Mealy type of finite state machine production of each state depends on present input and present state while in Moore type of state machine output of each only depend upon present state. Figure 1 shows the flow chart of mealy type of finite state machine while figure 2 shows the flow chart of Moore type of finite state machine [1].

As can be seen from Figure 1 in Mealy finite state machine combinatorial logic is to find the worth of next state depending upon the worth of input. Output is also determined by combinatorial logic base on input and present state value whereas Sequential logic is only used for storing states value. Same is the case in Figure 2 for Moore finite state machine but here output single depend upon the present state value [1].

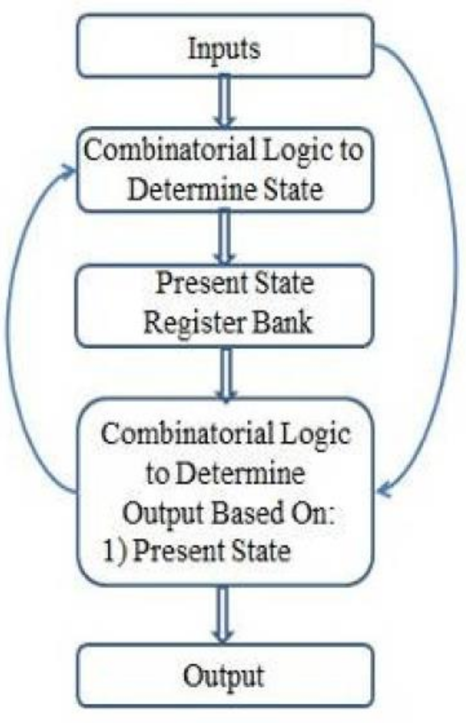
Inputs
Combinatorial Logic to Determine State
Present State Register Bank
Combinatorial Logic to Determine Output Based On: 1) Present State
Output

Fig. 2 Moore State Machine

To compare Mealy and Moore type of finite state machine one example is taken. Consider a circuit which is to spot a duo of 1's or 0's in the only bit input. If two one's or two zero's comes one after a new, output be supposed to go high. Or else output must be low. Here is a Moore type finite state machine evolution illustration for the circuit:

- When reset, state go to 00
- State will be 01, if input is 1,
- If input is 0, state go to 10
- State will be 11 if input repeats
- Later than state 11, goes to 10 states or 01 depending on the input. Once the state reaches the state 11 then allocate 1 to out since it is an asynchronous we have to make sure the reset earlier to making output high.

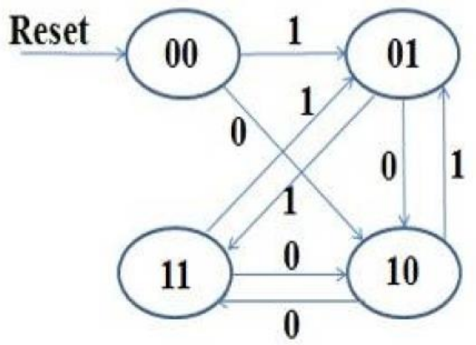

Fig. 3 State Diagram of Moore Machine

At present, let us design the above circuit by Mealy style state machine. Output of the design depends on equally state and input State evolution diagram is as follows:

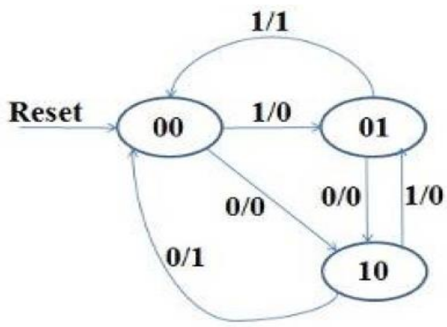

Fig. 4 State Diagram of Mealy Machine

III. Comparison Between Two State Machines

Moore type of finite state machine is easy to design as compare to Mealy type finite state machine. First design the states depending on the earlier state and input. Then intend output only depending on state. While in Mealy, both state and input are considered where designing the output. In Moore FSM four states are used to design the circuit and mealy FSM required only three states to design the same so there is more amount of states in mealy machine then Moore machine. As the result of it, amount of flip flops will be more in the Moore machine. In Mealy, output changes instantly when the input changes. We can detect this point in the simulation of the codes over. In Moore instance, output becomes high in the clock after that to the clock in which state goes 11. So, Mealy is faster than Moore. Mealy gives instant reaction to input and Moore gives response in the next clock. Now, most of the designer use only one always block to design state machine so combinational and sequential logic is mixed in this block so it may generate a problem when you manufacture the code and then authenticate gate level netlist. So It is more competent if you split combinational and chronological logic in the Mealy and Moore machines[1].

Modelling Mealy state machine

```
always@(in or pres_state) //Next State Decoder
always@(posedge clock) //Memory
always@(in or pres_state) //Output Decoder
```

Modelling Moore state machine

```
always@(in or pres_state) //NS Decoder
always@(posedge clock) //Memory
always@(pres_state) //Output Decoder
```

IV. FSM State Encoding Schemes

Each one state in a finite state machine can be represented with a only one of its kind pattern of one's and zeros and it is called state encoding. Two mainly trendy encoding schemes are binary and one hot encoding. In this article, will in brief talk about in cooperation of them and talk about how to choose the finest encoding scheme that suits your design, so proficient performance and source procedure can be ensured. In binary encoding the correlation connecting the number of state bits and number of states is represented by the following equation.

$$B = \log2(S)$$

So to execute the state machine with four states with a binary encoding scheme, two flip flops or state bits can be used to exceptionally encode four states as follows:

State1 = "00"
State2 = "01"
State3 = "10"
State4 = "11"

The Gray code binary programming scheme can also be used where one bit revolutionize at a time. Gray code binary scheme is helpful when the outputs of the state bits are used asynchronously. For example, if state machine switches from state '10' to '01' as it does in chronological binary programming and the registers do not switch the outputs as accurately the identical time , momentary outputs of either '11' or '00' can exist. This kind of instability can cause impulsive results throughout the circuit. A one-hot programming format uses one register for every state. For example four registers are used for a 4-state machine- by means of only one state bit high at a time. State programming can be done in this manner in one hot encoding:

State1 = "0001"
State2 = "0010"
State3 = "0100"
State4 = "1000"

V. Conclusion

In this article dissimilar types of state machines and types of programming schemes are discussed. Assessment among them is also completed. When there is require for more rapidly state machine then mealy machine is used but it makes intend multifaceted. In multifaceted intend, to make state machine intend simpler Moore machine is preferred. Programming schemes also play a vital role in seminal the area and recital of the state machine. Binary encoding scheme uses a smaller amount vicinity but can lead to impulsive recital if asynchronous productions are used. One hot encoding uses more area but has enhanced timing recital [1].

References

[1] Bhaumik Vaidy and Devani Anupam, "A Study of different Finite state machine design and efficient coding techniques,"Journal Of Information, Knowledge and Research in Electronics and Communication Engineering, Oct2013.
[2] IEEE standard Verilog Reference Manual , IEEE standard 2001
[3] Verilog FAQ by Shivkumar Chanod, Needamangalam Balachandar
[4] Design Compiler User Guide from Synopsys
[5] Verilog HDL by Samir Palnitkar